

IBE - Beyond Indistinguishability or Alphabetti Spaghetti

S.Srinivasan

RHUL
University of London

March 20, 2008/ PhD Student Seminar

Outline

- 1 Introduction
 - Provable Security
 - Identity Based Cryptography
 - Security Notions for IBE
- 2 Beyond Indistinguishability
 - Recipient Anonymity
 - The Multi-TA setting
 - The Multi-TA BasicIdent scheme
 - Security notions for m-IBE

What is Provable Security?

- Give a scheme
- Define a security notion
- Define capabilities of the adversary
- Make “precise” computational assumptions
- Specify the model
- Give a reduction

What is Provable Security?

- Give a scheme
- Define a security notion
- Define capabilities of the adversary
- Make "precise" computational assumptions
- Specify the model
- Give a reduction

What is Provable Security?

- Give a scheme
- Define a security notion
- Define capabilities of the adversary
- Make "precise" computational assumptions
- Specify the model
- Give a reduction

What is Provable Security?

- Give a scheme
- Define a security notion
- Define capabilities of the adversary
- Make “precise” computational assumptions
- Specify the model
- Give a reduction

What is Provable Security?

- Give a scheme
- Define a security notion
- Define capabilities of the adversary
- Make “precise” computational assumptions
- Specify the model
- Give a reduction

What is Provable Security?

- Give a scheme
- Define a security notion
- Define capabilities of the adversary
- Make “precise” computational assumptions
- Specify the model
- Give a reduction

An Example

- Scheme - Boneh-Franklin's BasicIdent IBE scheme
- Security notion - IND
- Capabilities of the adversary - CPA
- Computational assumptions - BDH is hard
- Random Oracle Model -ROM
- Reduction - BasicIdent is secure unless BDH is easy.

An Example

- Scheme - Boneh-Franklin's BasicIdent IBE scheme
- Security notion - IND
- Capabilities of the adversary - CPA
- Computational assumptions - BDH is hard
- Random Oracle Model -ROM
- Reduction - BasicIdent is secure unless BDH is easy.

An Example

- Scheme - Boneh-Franklin's BasicIdent IBE scheme
- Security notion - IND
- Capabilities of the adversary - CPA
- Computational assumptions - BDH is hard
- Random Oracle Model -ROM
- Reduction - BasicIdent is secure unless BDH is easy.

An Example

- Scheme - Boneh-Franklin's BasicIdent IBE scheme
- Security notion - IND
- Capabilities of the adversary - CPA
- Computational assumptions - BDH is hard
- Random Oracle Model -ROM
- Reduction - BasicIdent is secure unless BDH is easy.

An Example

- Scheme - Boneh-Franklin's BasicIdent IBE scheme
- Security notion - IND
- Capabilities of the adversary - CPA
- Computational assumptions - BDH is hard
- Random Oracle Model -ROM
- Reduction - BasicIdent is secure unless BDH is easy.

An Example

- Scheme - Boneh-Franklin's BasicIdent IBE scheme
- Security notion - IND
- Capabilities of the adversary - CPA
- Computational assumptions - BDH is hard
- Random Oracle Model -ROM
- Reduction - BasicIdent is secure unless BDH is easy.

Identity Based Cryptography

The concept of ID based Cryptography was introduced by Shamir in 1984.

- Alice uses Bob's ID (e-mail, phone number ..) as the Public Key
- Bob obtains his private key from Trent by proving he is the owner of the identity. This can happen **after he receives a message from Alice**. So communication can happen without the recipient setting up a key pair in advance.

Identity Based Cryptography

The concept of ID based Cryptography was introduced by Shamir in 1984.

- Alice uses Bob's ID (e-mail, phone number ..) as the Public Key
- Bob obtains his private key from Trent by proving he is the owner of the identity. This can happen **after he receives a message from Alice**. So communication can happen without the recipient setting up a key pair in advance.

IBE - pros and cons

- ID based cryptography can be used to secure communication where the recipient has not already set up a key pair.
- We can get around the problem of authenticating public keys and certificates.
- Need to obtain authentic TA parameters
- We still need a TA to verify the ownership of the identity.
- We need an authenticated and secure channel to deliver the private key!
- ID based schemes have inherent key escrow.
- What if TA is compromised?
- What about privacy/legislation?
- What about non-repudiation?

IBE - pros and cons

- ID based cryptography can be used to secure communication where the recipient has not already set up a key pair.
- We can get around the problem of authenticating public keys and certificates.
- Need to obtain authentic TA parameters
- We still need a TA to verify the ownership of the identity.
- We need an authenticated and secure channel to deliver the private key!
- ID based schemes have inherent key escrow.
- What if TA is compromised?
- What about privacy/legislation?
- What about non-repudiation?

IBE - pros and cons

- ID based cryptography can be used to secure communication where the recipient has not already set up a key pair.
- We can get around the problem of authenticating public keys and certificates.
- Need to obtain authentic TA parameters
 - We still need a TA to verify the ownership of the identity.
 - We need an authenticated and secure channel to deliver the private key!
- ID based schemes have inherent key escrow.
- What if TA is compromised?
- What about privacy/legislation?
- What about non-repudiation?

IBE - pros and cons

- ID based cryptography can be used to secure communication where the recipient has not already set up a key pair.
- We can get around the problem of authenticating public keys and certificates.
- Need to obtain authentic TA parameters
- We still need a TA to verify the ownership of the identity.
- We need an authenticated and secure channel to deliver the private key!
- ID based schemes have inherent key escrow.
- What if TA is compromised?
- What about privacy/legislation?
- What about non-repudiation?

IBE - pros and cons

- ID based cryptography can be used to secure communication where the recipient has not already set up a key pair.
- We can get around the problem of authenticating public keys and certificates.
- Need to obtain authentic TA parameters
- We still need a TA to verify the ownership of the identity.
- We need an authenticated and secure channel to deliver the private key!
- ID based schemes have inherent key escrow.
- What if TA is compromised?
- What about privacy/legislation?
- What about non-repudiation?

IBE - pros and cons

- ID based cryptography can be used to secure communication where the recipient has not already set up a key pair.
- We can get around the problem of authenticating public keys and certificates.
- Need to obtain authentic TA parameters
- We still need a TA to verify the ownership of the identity.
- We need an authenticated and secure channel to deliver the private key!
- ID based schemes have inherent key escrow.
 - What if TA is compromised?
 - What about privacy/legislation?
 - What about non-repudiation?

IBE - pros and cons

- ID based cryptography can be used to secure communication where the recipient has not already set up a key pair.
- We can get around the problem of authenticating public keys and certificates.
- Need to obtain authentic TA parameters
- We still need a TA to verify the ownership of the identity.
- We need an authenticated and secure channel to deliver the private key!
- ID based schemes have inherent key escrow.
- What if TA is compromised?
- What about privacy/legislation?
- What about non-repudiation?

IBE - pros and cons

- ID based cryptography can be used to secure communication where the recipient has not already set up a key pair.
- We can get around the problem of authenticating public keys and certificates.
- Need to obtain authentic TA parameters
- We still need a TA to verify the ownership of the identity.
- We need an authenticated and secure channel to deliver the private key!
- ID based schemes have inherent key escrow.
- What if TA is compromised?
- What about privacy/legislation?
- What about non-repudiation?

IBE - pros and cons

- ID based cryptography can be used to secure communication where the recipient has not already set up a key pair.
- We can get around the problem of authenticating public keys and certificates.
- Need to obtain authentic TA parameters
- We still need a TA to verify the ownership of the identity.
- We need an authenticated and secure channel to deliver the private key!
- ID based schemes have inherent key escrow.
- What if TA is compromised?
- What about privacy/legislation?
- What about non-repudiation?

Formally,

- **Setup**: On input 1^k , outputs a master public key mpk which includes system parameters $params$, and a master secret key msk . We assume the system parameters contain descriptions of the message and ciphertext space.
- **KeyDer**: A key derivation algorithm that on input mpk , msk and identifier $id \in \{0, 1\}^*$, returns a private key d_{id} .
- **Enc**: An encryption algorithm that on input mpk , identifier $id \in \{0, 1\}^*$ and message $m \in \text{MsgSp}$, returns a ciphertext c .
- **Dec**: A decryption algorithm that on input mpk , a private key d_{id} and a ciphertext c , returns either a message m or a failure symbol \perp .

Formally,

- **Setup**: On input 1^k , outputs a master public key mpk which includes system parameters $params$, and a master secret key msk . We assume the system parameters contain descriptions of the message and ciphertext space.
- **KeyDer**: A key derivation algorithm that on input mpk , msk and identifier $id \in \{0, 1\}^*$, returns a private key d_{id} .
- **Enc**: An encryption algorithm that on input mpk , identifier $id \in \{0, 1\}^*$ and message $m \in \text{MsgSp}$, returns a ciphertext c .
- **Dec**: A decryption algorithm that on input mpk , a private key d_{id} and a ciphertext c , returns either a message m or a failure symbol \perp .

Formally,

- **Setup**: On input 1^k , outputs a master public key mpk which includes system parameters $params$, and a master secret key msk . We assume the system parameters contain descriptions of the message and ciphertext space.
- **KeyDer**: A key derivation algorithm that on input mpk , msk and identifier $id \in \{0, 1\}^*$, returns a private key d_{id} .
- **Enc**: An encryption algorithm that on input mpk , identifier $id \in \{0, 1\}^*$ and message $m \in \text{MsgSp}$, returns a ciphertext c .
- **Dec**: A decryption algorithm that on input mpk , a private key d_{id} and a ciphertext c , returns either a message m or a failure symbol \perp .

Formally,

- **Setup**: On input 1^k , outputs a master public key mpk which includes system parameters $params$, and a master secret key msk . We assume the system parameters contain descriptions of the message and ciphertext space.
- **KeyDer**: A key derivation algorithm that on input mpk , msk and identifier $id \in \{0, 1\}^*$, returns a private key d_{id} .
- **Enc**: An encryption algorithm that on input mpk , identifier $id \in \{0, 1\}^*$ and message $m \in \text{MsgSp}$, returns a ciphertext c .
- **Dec**: A decryption algorithm that on input mpk , a private key d_{id} and a ciphertext c , returns either a message m or a failure symbol \perp .

These algorithms must satisfy the standard consistency requirement that Decryption undoes Encryption. i.e

$$\forall m \in \mathcal{M} : \text{Dec}(mpk, c, d_{id}) = m \text{ where } c = \text{Enc}(mpk, m, id)$$

IND-CCA

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{IND-CCA-b}}(k)$

$IDSet \leftarrow \emptyset, CSet \leftarrow \emptyset$

$(mpk, msk) \leftarrow \text{Setup}(1^k)$

$(id, m_0, m_1, state) \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{find}, mpk)$

$c^* \leftarrow \text{Enc}(mpk, id, m_b)$

$b' \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{guess}, c^*, state)$

If $\{m_0, m_1\} \not\subseteq \text{MsgSp}(k)$ or $|m_0| \neq |m_1|$ then return 0

If $id \notin IDSet$ and $\{id, c^*\} \not\subseteq CSet$ then return b' else return 0.

Oracles

Oracle $\text{KeyDer}(id)$
 $IDSet \leftarrow IDSet \cup \{id\}$
 $usk[id] \leftarrow \text{KeyDer}(msk, id)$
Return $usk[id]$

Oracle $\text{Dec}(id, c)$
 $CSet \leftarrow CSet \cup \{id, c\}$
 $m \leftarrow \text{Dec}(msk, id, c)$
Return m

Advantage of the Adversary

The advantage of \mathcal{A} against the IBE scheme in the above IND-CCA security game is defined to be:

$$\mathbf{Adv}_{\mathcal{A}}^{\text{IND-CCA}}(k) = \left| \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{IND-CCA-1}}(k) = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{IND-CCA-0}}(k) = 1] \right|$$

An IBE scheme is secure if all IND-CCA adversaries have **negligible** advantage.

Negligible?

Definition

A function $\epsilon(k)$ is said to be *negligible* if, for every c , there exists k_c such that $\epsilon(k) \leq k^{-c}$ for every $k \geq k_c$.

IND-CPA

Removing the adversary's access to the Decryption Oracle gives the IND-CPA security notion with the advantage defined in a similar fashion.

- So is IND-CCA the best we can get (or need) ?

Recipient Anonymity

- Ciphertext should not leak identity of the recipient
- (A step towards?) Fully private communications
- RA-CPA for PEKS

Recipient Anonymity

- Ciphertext should not leak identity of the recipient
- (A step towards?) Fully private communications
- RA-CPA for PEKS

Recipient Anonymity

- Ciphertext should not leak identity of the recipient
- (A step towards?) Fully private communications
- RA-CPA for PEKS

Recipient Anonymity

Experiment $\text{Exp}_{\mathcal{A}}^{\text{RA-CCA-b}}(k)$

$IDSet \leftarrow \emptyset, CSet \leftarrow \emptyset$

$(mpk, msk) \leftarrow \text{Setup}(1^k)$

$(id_0, id_1, m, state) \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{find}, mpk)$

$c^* \leftarrow \text{Enc}(mpk, id_b, m)$

$b' \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{guess}, c^*, state)$

If $m \notin \text{MsgSp}(k)$ then return 0

If $\{id_0, id_1\} \not\subseteq IDSet$ and

$\{id_0, c^*\} \not\subseteq CSet$ and $\{id_1, c^*\} \not\subseteq CSet$ then

return b' else return 0.

- So are we done?

The Multi-TA setting

- A set of TA's may share common parameters (with different public keys of course)

An Example

- Quick Overview of Pairings
- Show the Multi-TA BasicIdent scheme as an example

An Example

- Quick Overview of Pairings
- Show the Multi-TA BasicIdent scheme as an example

Introduction to Pairings

Given G_1 , G_2 , additive groups of prime order q and G_T , a multiplicative group of prime order q . Let P be an element of G_1 . Since the groups have prime order, P is a generator of G_1 . (Likewise for Q in G_2).

A pairing is map $\hat{e} : G_1 \times G_2 \rightarrow G_T$ such that it is

- **Bilinear**
 - $\forall P \in G_1$ and $\forall Q \in G_2$ and $a, b \in \mathbb{Z}$
 - $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
- **Non Degenerate**
 - $\hat{e}(P_1, P_2) \neq 1$
- **Efficiently Computable**

If the two inputs are in the same group, we call it a symmetric pairing.

A pairing is map $\hat{e} : G_1 \times G_2 \rightarrow G_T$ such that it is

- **Bilinear**
- $\forall P \in G_1$ and $\forall Q \in G_2$ and $a, b \in \mathbb{Z}$
 - $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
- **Non Degenerate**
- $\hat{e}(P_1, P_2) \neq 1$
- **Efficiently Computable**

If the two inputs are in the same group, we call it a symmetric pairing.

A pairing is map $\hat{e} : G_1 \times G_2 \rightarrow G_T$ such that it is

- **Bilinear**
- $\forall P \in G_1$ and $\forall Q \in G_2$ and $a, b \in \mathbb{Z}$
- $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
- **Non Degenerate**
- $\hat{e}(P_1, P_2) \neq 1$
- **Efficiently Computable**

If the two inputs are in the same group, we call it a symmetric pairing.

A pairing is map $\hat{e} : G_1 \times G_2 \rightarrow G_T$ such that it is

- **Bilinear**
- $\forall P \in G_1$ and $\forall Q \in G_2$ and $a, b \in \mathbb{Z}$
- $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
- **Non Degenerate**
- $\hat{e}(P_1, P_2) \neq 1$
- **Efficiently Computable**

If the two inputs are in the same group, we call it a symmetric pairing.

A pairing is map $\hat{e} : G_1 \times G_2 \rightarrow G_T$ such that it is

- **Bilinear**
- $\forall P \in G_1$ and $\forall Q \in G_2$ and $a, b \in \mathbb{Z}$
- $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
- **Non Degenerate**
- $\hat{e}(P_1, P_2) \neq 1$
- **Efficiently Computable**

If the two inputs are in the same group, we call it a symmetric pairing.

A pairing is map $\hat{e} : G_1 \times G_2 \rightarrow G_T$ such that it is

- **Bilinear**
- $\forall P \in G_1$ and $\forall Q \in G_2$ and $a, b \in \mathbb{Z}$
- $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
- **Non Degenerate**
- $\hat{e}(P_1, P_2) \neq 1$
- **Efficiently Computable**

If the two inputs are in the same group, we call it a symmetric pairing.

Definition

A pairing-friendly group generator PairingGen is a polynomial time algorithm with input 1^k and output a tuple (G, G_T, e, q, P) . Here G, G_T are groups of prime order q , P generates G , and $e : G \times G \rightarrow G_T$ is a bilinear, non-degenerate and efficiently computable map. By convention, G is an additive group and G_T multiplicative.

Pairing based hard problems

- The Bilinear Diffie Hellman Problem (BDHP).
Given elements $P, Q = aP, R = bP, S = cP \in G_1$,
 $a, b, c \in \mathbb{Z}_q^*$, unknown, finding $\hat{e}(P, P)^{abc}$ is hard.
- The hardness of the BDHP can be related to the more well understood and familiar hard problems like the Computational Diffie Hellman problem (CDHP) and the Discrete Log Problem (DLP). We can show that the BDHP is no harder than the CDHP or the DLP in G_1 .

Pairing based hard problems

- The Bilinear Diffie Hellman Problem (BDHP).
Given elements $P, Q = aP, R = bP, S = cP \in G_1$,
 $a, b, c \in \mathbb{Z}_q^*$, unknown, finding $\hat{e}(P, P)^{abc}$ is hard.
- The hardness of the BDHP can be related to the more well understood and familiar hard problems like the Computational Diffie Hellman problem (CDHP) and the Discrete Log Problem (DLP). We can show that the BDHP is no harder than the CDHP or the DLP in G_1 .

The Multi-TA Boneh-Franklin BasicIdent IBE

CommonSetup(1^k):

- $(G, G_T, e, q, P) \leftarrow \text{PairingGen}(1^k)$.
- Output $params = (G, G_T, e, q, P, H_1, H_2, n)$ where $H_1 : \{0, 1\}^* \rightarrow G, H_2 : G_T \rightarrow \{0, 1\}^n$ for some $n = n(k)$.
- $\text{MsgSp} = \{0, 1\}^n, \text{CtSp} = G_1 \times \{0, 1\}^n, \text{RSp} = \mathbb{Z}_q$.

TASetup($params$)

- Set $s \leftarrow \mathbb{Z}_q$.
- Set $mpk = (params, sP)$.
- Set $msk = s$.
- Output (mpk, msk) .

The Multi-TA Boneh-Franklin BasicIdent IBE

CommonSetup(1^k):

- $(G, G_T, e, q, P) \leftarrow \text{PairingGen}(1^k)$.
- Output $params = (G, G_T, e, q, P, H_1, H_2, n)$ where $H_1 : \{0, 1\}^* \rightarrow G$, $H_2 : G_T \rightarrow \{0, 1\}^n$ for some $n = n(k)$.
- $\text{MsgSp} = \{0, 1\}^n$, $\text{CtSp} = G_1 \times \{0, 1\}^n$, $\text{RSp} = \mathbb{Z}_q$.

TASetup($params$)

- Set $s \leftarrow \mathbb{Z}_q$.
- Set $mpk = (params, sP)$.
- Set $msk = s$.
- Output (mpk, msk) .

The Multi-TA Boneh-Franklin BasicIdent IBE

CommonSetup(1^k):

- $(G, G_T, e, q, P) \leftarrow \text{PairingGen}(1^k)$.
- Output $params = (G, G_T, e, q, P, H_1, H_2, n)$ where $H_1 : \{0, 1\}^* \rightarrow G$, $H_2 : G_T \rightarrow \{0, 1\}^n$ for some $n = n(k)$.
- $\text{MsgSp} = \{0, 1\}^n$, $\text{CtSp} = G_1 \times \{0, 1\}^n$, $\text{RSp} = \mathbb{Z}_q$.

TASetup($params$)

- Set $s \leftarrow \mathbb{Z}_q$.
- Set $mpk = (params, sP)$.
- Set $msk = s$.
- Output (mpk, msk) .

The Multi-TA Boneh-Franklin BasicIdent IBE

CommonSetup(1^k):

- $(G, G_T, e, q, P) \leftarrow \text{PairingGen}(1^k)$.
- Output $params = (G, G_T, e, q, P, H_1, H_2, n)$ where $H_1 : \{0, 1\}^* \rightarrow G$, $H_2 : G_T \rightarrow \{0, 1\}^n$ for some $n = n(k)$.
- $\text{MsgSp} = \{0, 1\}^n$, $\text{CtSp} = G_1 \times \{0, 1\}^n$, $\text{RSp} = \mathbb{Z}_q$.

TASetup($params$)

- Set $s \leftarrow \mathbb{Z}_q$.
- Set $mpk = (params, sP)$.
- Set $msk = s$.
- Output (mpk, msk) .

The Multi-TA Boneh-Franklin BasicIdent IBE

CommonSetup(1^k):

- $(G, G_T, e, q, P) \leftarrow \text{PairingGen}(1^k)$.
- Output $params = (G, G_T, e, q, P, H_1, H_2, n)$ where $H_1 : \{0, 1\}^* \rightarrow G$, $H_2 : G_T \rightarrow \{0, 1\}^n$ for some $n = n(k)$.
- $\text{MsgSp} = \{0, 1\}^n$, $\text{CtSp} = G_1 \times \{0, 1\}^n$, $\text{RSp} = \mathbb{Z}_q$.

TASetup($params$)

- Set $s \leftarrow \mathbb{Z}_q$.
- Set $mpk = (params, sP)$.
- Set $msk = s$.
- Output (mpk, msk) .

The Multi-TA Boneh-Franklin BasicIdent IBE

CommonSetup(1^k):

- $(G, G_T, e, q, P) \leftarrow \text{PairingGen}(1^k)$.
- Output $params = (G, G_T, e, q, P, H_1, H_2, n)$ where $H_1 : \{0, 1\}^* \rightarrow G$, $H_2 : G_T \rightarrow \{0, 1\}^n$ for some $n = n(k)$.
- $\text{MsgSp} = \{0, 1\}^n$, $\text{CtSp} = G_1 \times \{0, 1\}^n$, $\text{RSp} = \mathbb{Z}_q$.

TASetup($params$)

- Set $s \leftarrow \mathbb{Z}_q$.
- Set $mpk = (params, sP)$.
- Set $msk = s$.
- Output (mpk, msk) .

The Multi-TA Boneh-Franklin BasicIdent IBE

CommonSetup(1^k):

- $(G, G_T, e, q, P) \leftarrow \text{PairingGen}(1^k)$.
- Output $params = (G, G_T, e, q, P, H_1, H_2, n)$ where $H_1 : \{0, 1\}^* \rightarrow G$, $H_2 : G_T \rightarrow \{0, 1\}^n$ for some $n = n(k)$.
- $\text{MsgSp} = \{0, 1\}^n$, $\text{CtSp} = G_1 \times \{0, 1\}^n$, $\text{RSp} = \mathbb{Z}_q$.

TASetup($params$)

- Set $s \leftarrow \mathbb{Z}_q$.
- Set $mpk = (params, sP)$.
- Set $msk = s$.
- Output (mpk, msk) .

KeyDer^{H₁}(*ta*, *id*):

- Output $usk_{id,ta} = msk_{ta} \cdot H_1(id)$.

Enc^{H₁,H₂}(*ta*, *id*, *m*):

- Set $r \xleftarrow{\$} \mathbb{Z}_q$.
- Set $T = e(H_1(id), mpk_{ta})^r$.
- Output $c = (rP, m \oplus H_2(T))$.

Dec^{H₂}(*ta*, $usk_{id,ta}$, *c*):

- Parse *c* as (*U*, *V*).
- Set $T = e(usk_{id,ta}, U)$.
- Output $m = V \oplus H_2(T)$.

KeyDer^{H₁}(*ta*, *id*):

- Output $usk_{id,ta} = msk_{ta} \cdot H_1(id)$.

Enc^{H₁,H₂}(*ta*, *id*, *m*):

- Set $r \xleftarrow{\$} \mathbb{Z}_q$.
- Set $T = e(H_1(id), mpk_{ta})^r$.
- Output $c = (rP, m \oplus H_2(T))$.

Dec^{H₂}(*ta*, $usk_{id,ta}$, *c*):

- Parse *c* as (*U*, *V*).
- Set $T = e(usk_{id,ta}, U)$.
- Output $m = V \oplus H_2(T)$.

KeyDer^{H₁}(*ta*, *id*):

- Output $usk_{id,ta} = msk_{ta} \cdot H_1(id)$.

Enc^{H₁,H₂}(*ta*, *id*, *m*):

- Set $r \xleftarrow{\$} \mathbb{Z}_q$.
- Set $T = e(H_1(id), mpk_{ta})^r$.
- Output $c = (rP, m \oplus H_2(T))$.

Dec^{H₂}(*ta*, $usk_{id,ta}$, *c*):

- Parse *c* as (*U*, *V*).
- Set $T = e(usk_{id,ta}, U)$.
- Output $m = V \oplus H_2(T)$.

KeyDer^{H₁}(*ta*, *id*):

- Output $usk_{id,ta} = msk_{ta} \cdot H_1(id)$.

Enc^{H₁,H₂}(*ta*, *id*, *m*):

- Set $r \xleftarrow{\$} \mathbb{Z}_q$.
- Set $T = e(H_1(id), mpk_{ta})^r$.
- Output $c = (rP, m \oplus H_2(T))$.

Dec^{H₂}(*ta*, $usk_{id,ta}$, *c*):

- Parse *c* as (*U*, *V*).
- Set $T = e(usk_{id,ta}, U)$.
- Output $m = V \oplus H_2(T)$.

KeyDer^{H₁}(*ta*, *id*):

- Output $usk_{id,ta} = msk_{ta} \cdot H_1(id)$.

Enc^{H₁,H₂}(*ta*, *id*, *m*):

- Set $r \xleftarrow{\$} \mathbb{Z}_q$.
- Set $T = e(H_1(id), mpk_{ta})^r$.
- Output $c = (rP, m \oplus H_2(T))$.

Dec^{H₂}(*ta*, $usk_{id,ta}$, *c*):

- Parse *c* as (*U*, *V*).
- Set $T = e(usk_{id,ta}, U)$.
- Output $m = V \oplus H_2(T)$.

KeyDer^{H₁}(*ta*, *id*):

- Output $usk_{id,ta} = msk_{ta} \cdot H_1(id)$.

Enc^{H₁,H₂}(*ta*, *id*, *m*):

- Set $r \xleftarrow{\$} \mathbb{Z}_q$.
- Set $T = e(H_1(id), mpk_{ta})^r$.
- Output $c = (rP, m \oplus H_2(T))$.

Dec^{H₂}(*ta*, $usk_{id,ta}$, *c*):

- Parse *c* as (*U*, *V*).
- Set $T = e(usk_{id,ta}, U)$.
- Output $m = V \oplus H_2(T)$.

KeyDer^{H₁}(*ta*, *id*):

- Output $usk_{id,ta} = msk_{ta} \cdot H_1(id)$.

Enc^{H₁,H₂}(*ta*, *id*, *m*):

- Set $r \xleftarrow{\$} \mathbb{Z}_q$.
- Set $T = e(H_1(id), mpk_{ta})^r$.
- Output $c = (rP, m \oplus H_2(T))$.

Dec^{H₂}(*ta*, $usk_{id,ta}$, *c*):

- Parse *c* as (*U*, *V*).
- Set $T = e(usk_{id,ta}, U)$.
- Output $m = V \oplus H_2(T)$.

m-IND-CCA

Experiment $\text{Exp}_{\mathcal{A}}^{\text{m-IND-CCA-b}}(k)$

$params \leftarrow \text{CommonSetup}(1^k)$

$\forall ta \in \mathcal{TA}, (mpk_{ta}, msk_{ta}) \leftarrow \text{TASetup}(params),$

$IDSet_{ta} \leftarrow \emptyset$ and $CSet_{ta} \leftarrow \emptyset$

$(id, ta, m_0, m_1, state) \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{find}, MPK)$

$c^* \leftarrow \text{Enc}(mpk_{ta}, id, m_b)$

$b' \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{guess}, c^*, state)$

If $\{m_0, m_1\} \not\subseteq \text{MsgSp}(k)$ or $|m_0| \neq |m_1|$ then return 0

If $id \notin IDSet_{ta}$ and $(c^*, id) \notin CSet_{ta}$ then return b' else return 0

Oracles

Oracle $\text{KeyDer}(id, ta)$
 $IDSet_{ta} \leftarrow IDSet_{ta} \cup \{id\}$
 $usk_{id,ta} \leftarrow \text{KeyDer}(msk_{ta}, id)$
Return $usk_{id,ta}$

Oracle $\text{Dec}(id, ta, c)$
 $CSet_{ta} \leftarrow CSet_{ta} \cup (c, id)$
 $usk_{id,ta} \leftarrow \text{KeyDer}(msk_{ta}, id)$
 $m \leftarrow \text{Dec}(mpk_{ta}, usk_{id,ta}, c)$
Return m

Theorem

Let $\text{atk} \in \{\text{CPA}, \text{CCA}\}$. Then for any m -IND- atk adversary \mathcal{A} against a multi-TA IBE scheme with n TAs having advantage ε and running in time t , there exists an IND- atk adversary \mathcal{B} against the corresponding single-TA IBE scheme with advantage $\frac{\varepsilon}{n}$ and running in time $O(\text{time}(\mathcal{A}))$.

m-RA-CCA

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{m-RA-CCA-b}}(k)$
 $params \leftarrow \text{CommonSetup}(1^k)$
 $\forall ta \in \mathcal{TA}, (mpk_{ta}, msk_{ta}) \leftarrow \text{TASetup}(params),$
 $IDSet_{ta} \leftarrow \emptyset$ and $CSet_{ta} \leftarrow \emptyset$
 $(id_0, id_1, ta, m, state) \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{find}, MPK)$
 $c^* \leftarrow \text{Enc}(mpk_{ta}, id_b, m)$
 $b' \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{guess}, c^*, state)$
 If $m \notin \text{MsgSp}(k)$ then return 0
 If $id_0 \notin IDSet_{ta}, id_1 \notin IDSet_{ta}, (c^*, id_0) \notin CSet_{ta}$
 and $(c^*, id_1) \notin CSet_{ta}$ then return b' else return 0

Theorem

Let $atk \in \{CPA, CCA\}$. Then for any m -RA- atk adversary \mathcal{A} against a multi-TA IBE scheme with n TAs having advantage ε and running in time t , there exists an RA- atk adversary \mathcal{B} against the corresponding single-TA IBE scheme with advantage $\frac{\varepsilon}{n}$ and running in time $O(\text{time}(\mathcal{A}))$.

m-RA-RE-CCA

Experiment $\text{Exp}_{\mathcal{A}}^{\text{m-RA-RE-CCA-b}}(k)$
 $params \leftarrow \text{CommonSetup}(1^k)$
 $\forall ta \in \mathcal{TA}, (mpk_{ta}, msk_{ta}) \leftarrow \text{TASetup}(params),$
 $IDSet_{ta} \leftarrow \emptyset$ and $CSet_{ta} \leftarrow \emptyset$
 $(id_0, id_1, ta, m, state) \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{find}, MPK)$
 $m' \xleftarrow{\$} \{0, 1\}^{|m|}; c^* \leftarrow \text{Enc}(mpk_{ta}, id_b, m')$
 $b' \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{guess}, c^*, state)$
 If $m \notin \text{MsgSp}(k)$ then return 0
 If $id_0 \notin IDSet_{ta}, id_1 \notin IDSet_{ta}, (c^*, id_0) \notin CSet_{ta}$
 and $(c^*, id_1) \notin CSet_{ta}$ then return b' else return 0

Lemma

Let m -IBE be a multi-TA IBE scheme that is m -IND- atk and m -RA-RE- atk secure. Then it is also m -RA- atk secure. Here $atk \in \{CPA, CCA\}$

m-TAA-CCA

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{m-TAA-CCA-b}}(k)$
 $params \leftarrow \text{CommonSetup}(1^k)$
 $\forall ta \in \mathcal{TA}, (mpk_{ta}, msk_{ta}) \leftarrow \text{TASetup}(params),$
 $IDSet_{ta} \leftarrow \emptyset$ and $CSet_{ta} \leftarrow \emptyset$
 $(id, m, ta_0, ta_1, state) \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{find}, MPK)$
 $c^* \leftarrow \text{Enc}(mpk_{ta_b}, id, m)$
 $b' \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{guess}, c^*, state)$
 If $m \notin \text{MsgSp}(k)$ then return 0
 If $id \notin IDSet_{ta_0}, id \notin IDSet_{ta_1}, (c^*, id) \notin CSet_{ta_0}$
 and $(c^*, id) \notin CSet_{ta_1}$ then return b' else return 0

m-TAA-RE-CCA

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{m-TAA-CCA-b}}(k)$
 $params \leftarrow \text{CommonSetup}(1^k)$
 $\forall ta \in \mathcal{TA}, (mpk_{ta}, msk_{ta}) \leftarrow \text{TASetup}(params),$
 $IDSet_{ta} \leftarrow \emptyset$ and $CSet_{ta} \leftarrow \emptyset$
 $(id, m, ta_0, ta_1, state) \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{find}, MPK)$
 $m' \xrightarrow{\$} \{0, 1\}^{|m|}; c^* \leftarrow \text{Enc}(mpk_{ta_b}, id, m')$
 $b' \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{guess}, c^*, state)$
 If $m \notin \text{MsgSp}(k)$ then return 0
 If $id \notin IDSet_{ta_0}, id \notin IDSet_{ta_1}, (c^*, id) \notin CSet_{ta_0}$
 and $(c^*, id) \notin CSet_{ta_1}$ then return b' else return 0.

Lemma

Let m -IBE be a multi-TA IBE scheme that is m -IND- atk and m -TAA-RE- atk secure. Then it is also m -TAA- atk secure. Here $atk \in \{CPA, CCA\}$.

m-IND-RA-TAA-CCA

Experiment $\text{Exp}_{\mathcal{A}}^{\text{m-IND-RA-TAA-CCA-b}}(k)$

$params \leftarrow \text{CommonSetup}(1^k)$

$\forall ta \in \mathcal{TA}, (mpk_{ta}, msk_{ta}) \leftarrow \text{TASetup}(params),$

$IDSet_{ta} \leftarrow \emptyset$ and $CSet_{ta} \leftarrow \emptyset$

$(id_0, id_1, m_0, m_1, ta_0, ta_1, state) \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{find}, MPK)$

$c^* \leftarrow \text{Enc}(mpk_{ta_b}, id_b, m_b)$

$b' \leftarrow \mathcal{A}^{\text{KeyDer,Dec}}(\text{guess}, c^*, state)$

If $\{m_0, m_1\} \not\subseteq \text{MsgSp}(k)$ or $|m_0| \neq |m_1|$ then return 0

If $id_0 \notin IDSet_{ta_0}, id_1 \notin IDSet_{ta_1}, (c^*, id_0) \notin CSet_{ta_0}$
 and $(c^*, id_1) \notin CSet_{ta_1}$ then return b' else return 0.

Lemma

Let IBE be an IBE scheme that is m -IND- atk , m -RA- atk and m -TAA- atk secure. Then it is also m -IND-RA-TAA- atk secure. Here $atk \in \{CPA, CCA\}$.

- using all the above results and a modified Fujisaki Okamoto transformation for IBE (not covered here), we prove

Theorem

Applying the modified Fujisaki Okamoto Transformation to the Multi-TA BasicIdent scheme gives a scheme that is m -IND-RA-TAA-CCA.

..backwards understood be only can but, forward lived be to has Life

SUMMARY

- Blitzed through Provable Security, IBE, Pairings
- Motivated Multi-TA IBE
- Showed a multi-TA instantiation based on BasicIdent
- Gave new security notions for Multi-TA IBE and relations between them
- Stated some results

..backwards understood be only can but, forward lived be to has Life

SUMMARY

- Blitized through Provable Security, IBE, Pairings
- Motivated Multi-TA IBE
- Showed a multi-TA instantiation based on BasicIdent
- Gave new security notions for Multi-TA IBE and relations between them
- Stated some results

..backwards understood be only can but, forward lived be to has Life

SUMMARY

- Blitzed through Provable Security, IBE, Pairings
- Motivated Multi-TA IBE
- Showed a multi-TA instantiation based on BasicIdent
- Gave new security notions for Multi-TA IBE and relations between them
- Stated some results

..backwards understood be only can but, forward lived be to has Life

SUMMARY

- Blitzed through Provable Security, IBE, Pairings
- Motivated Multi-TA IBE
- Showed a multi-TA instantiation based on BasicIdent
- Gave new security notions for Multi-TA IBE and relations between them
- Stated some results

..backwards understood be only can but, forward lived be to has Life

SUMMARY

- Blitzed through Provable Security, IBE, Pairings
- Motivated Multi-TA IBE
- Showed a multi-TA instantiation based on BasicIdent
- Gave new security notions for Multi-TA IBE and relations between them
- Stated some results

- All work is joint work with my supervisor Kenny Paterson.
- These slides will be available on
- <http://www.isg.rhul.ac.uk/~prai175>
- <http://www.isg.rhul.ac.uk/~prai175/ISGStudentSem07/index.html>

- All work is joint work with my supervisor Kenny Paterson.
- These slides will be available on
- <http://www.isg.rhul.ac.uk/~prai175>
- [http://www.isg.rhul.ac.uk/~prai175/
ISGStudentSem07/index.html](http://www.isg.rhul.ac.uk/~prai175/ISGStudentSem07/index.html)

- All work is joint work with my supervisor Kenny Paterson.
- These slides will be available on
- <http://www.isg.rhul.ac.uk/~prai175>
- [http://www.isg.rhul.ac.uk/~prai175/
ISGStudentSem07/index.html](http://www.isg.rhul.ac.uk/~prai175/ISGStudentSem07/index.html)

- All work is joint work with my supervisor Kenny Paterson.
- These slides will be available on
- <http://www.isg.rhul.ac.uk/~prai175>
- [http://www.isg.rhul.ac.uk/~prai175/
ISGStudentSem07/index.html](http://www.isg.rhul.ac.uk/~prai175/ISGStudentSem07/index.html)

- Thanks for coming! Have a nice Easter Break!